Selected Data Manipulation Techniques in SAS

Using SAS Studio on SAS On Demand for Academics (SODA)

Imported Your Data Already?

- If you already have your data in SAS Studio on SAS On Demand for Academics, you can skip the slides providing an overview of the import process.

Dataset

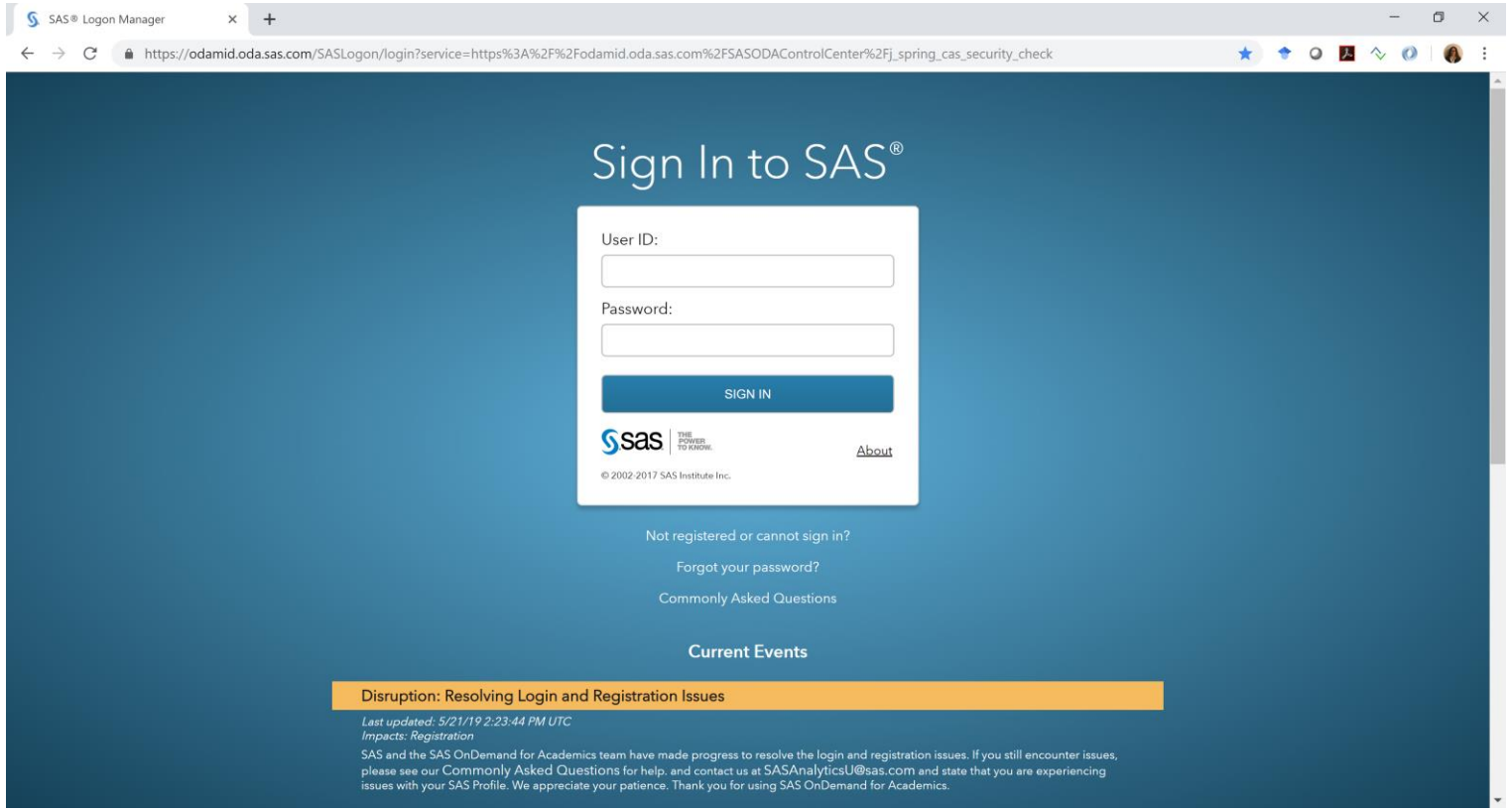
- This tutorial is a walkthrough with a sample set of data. You may use this to walk through the tutorial, if you wish, but for your assignments, you will be asked to use your own dataset (as specified within the course).

Dataset reference:

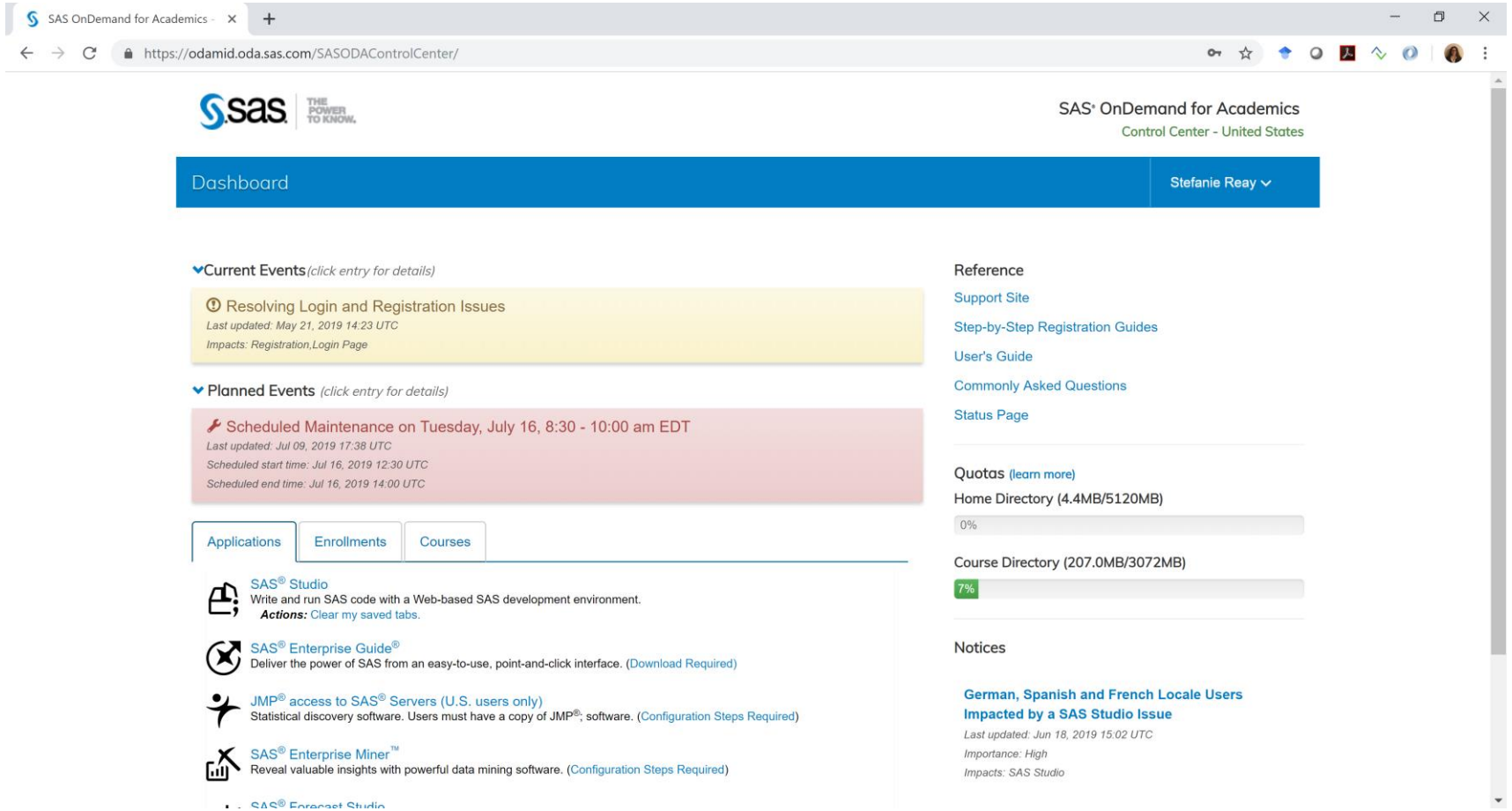
Skoryk, M. (2021). Sepsis Prediction from Clinical Data. Version 1. Retrieved from https://www.kaggle.com/maxskoryk/datasepsis

Access the SAS OnDemand for Academics Control Center

https://odamid.oda.sas.com/SASODAControlCenter

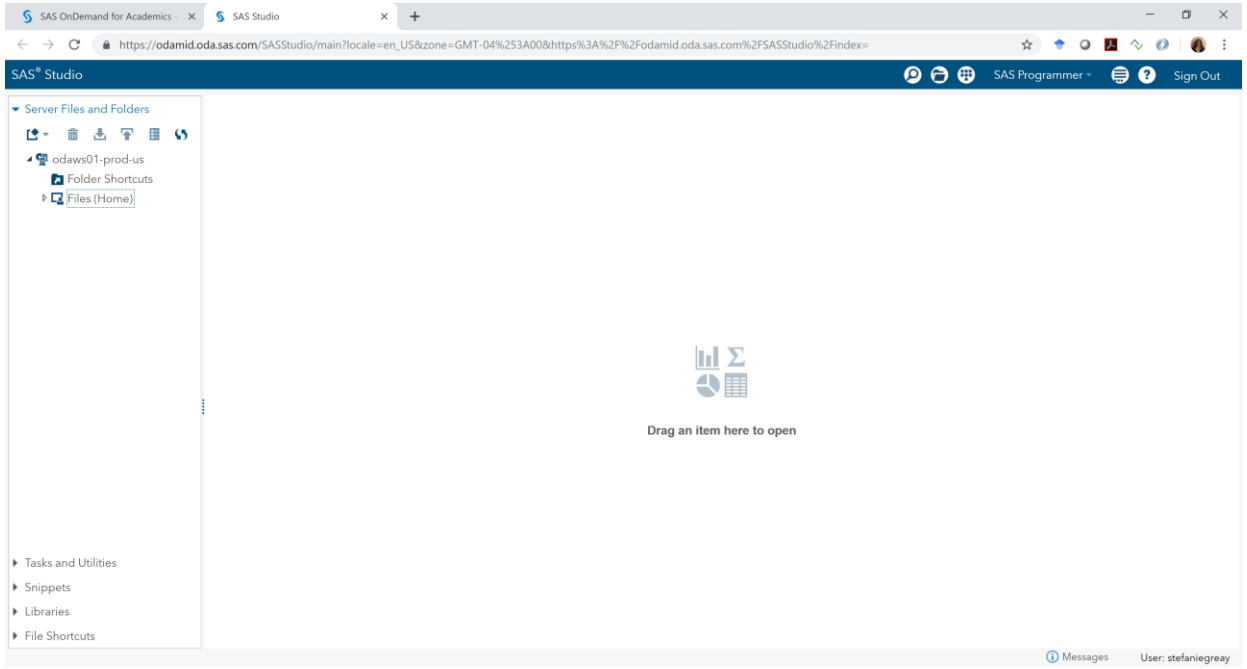


SAS OnDemand for Academics (SODA) Control Center

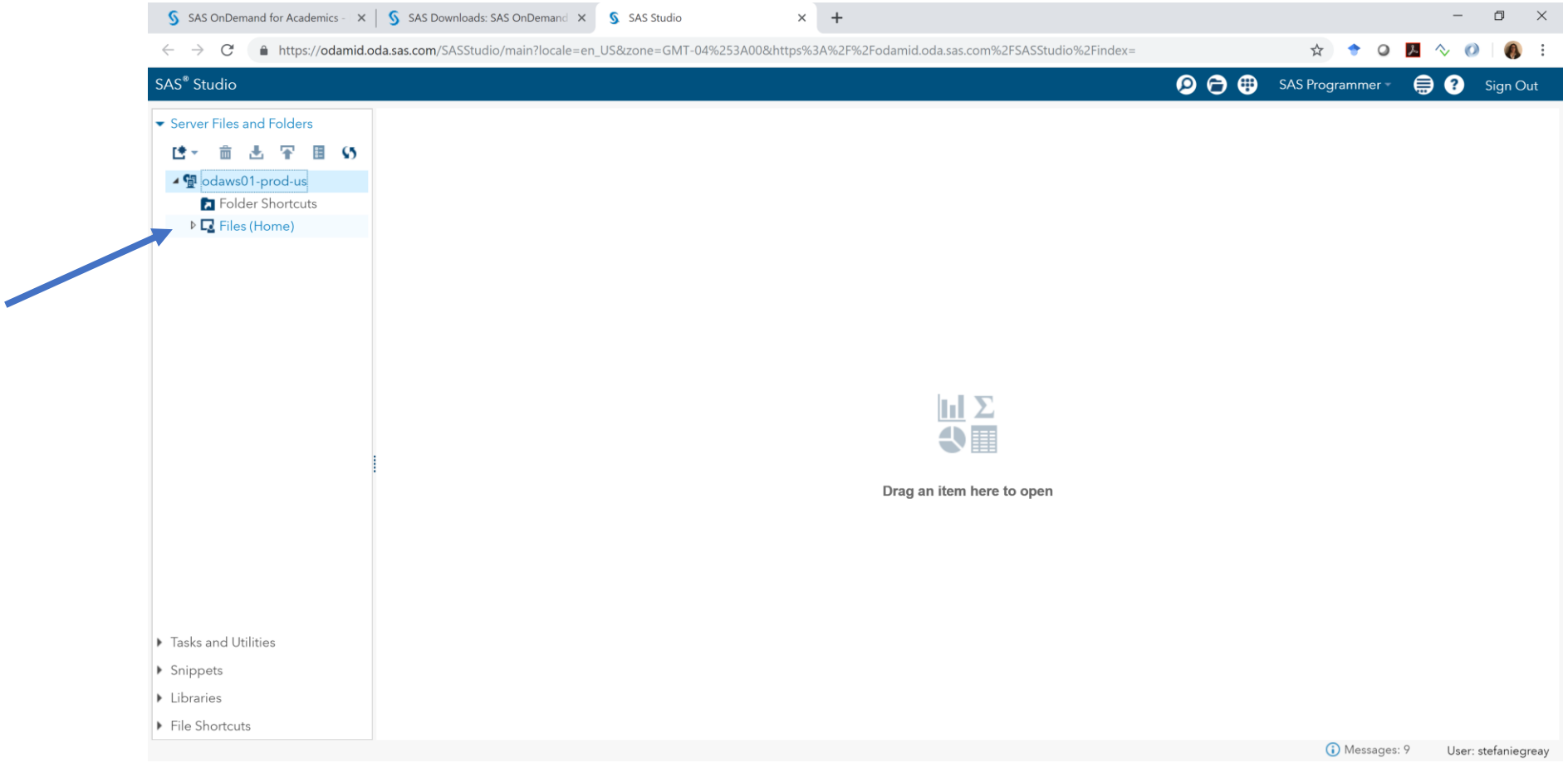


SAS Studio

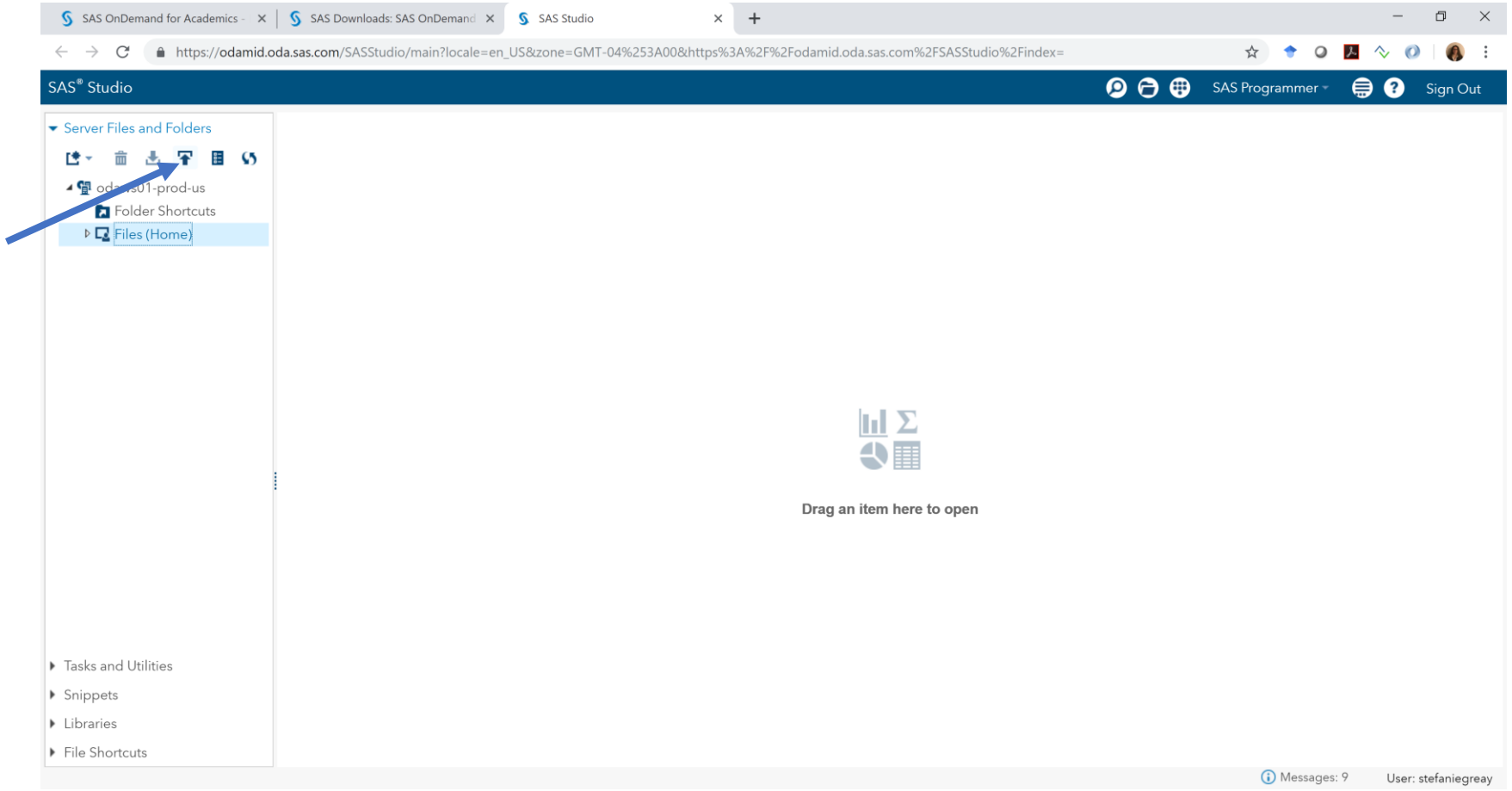
https://odamid.oda.sas.com/SASStudio/



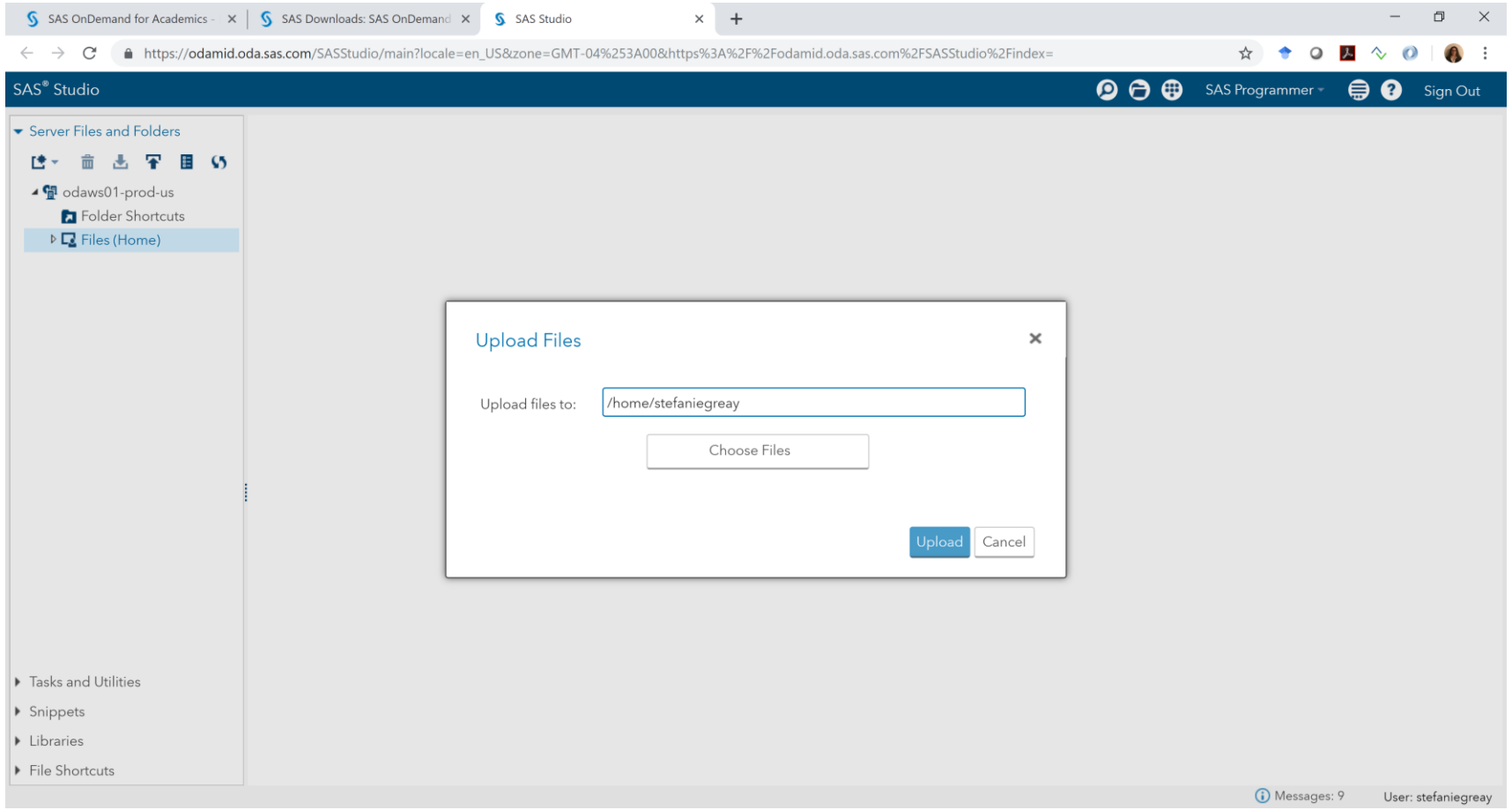
Click on Files(Home)



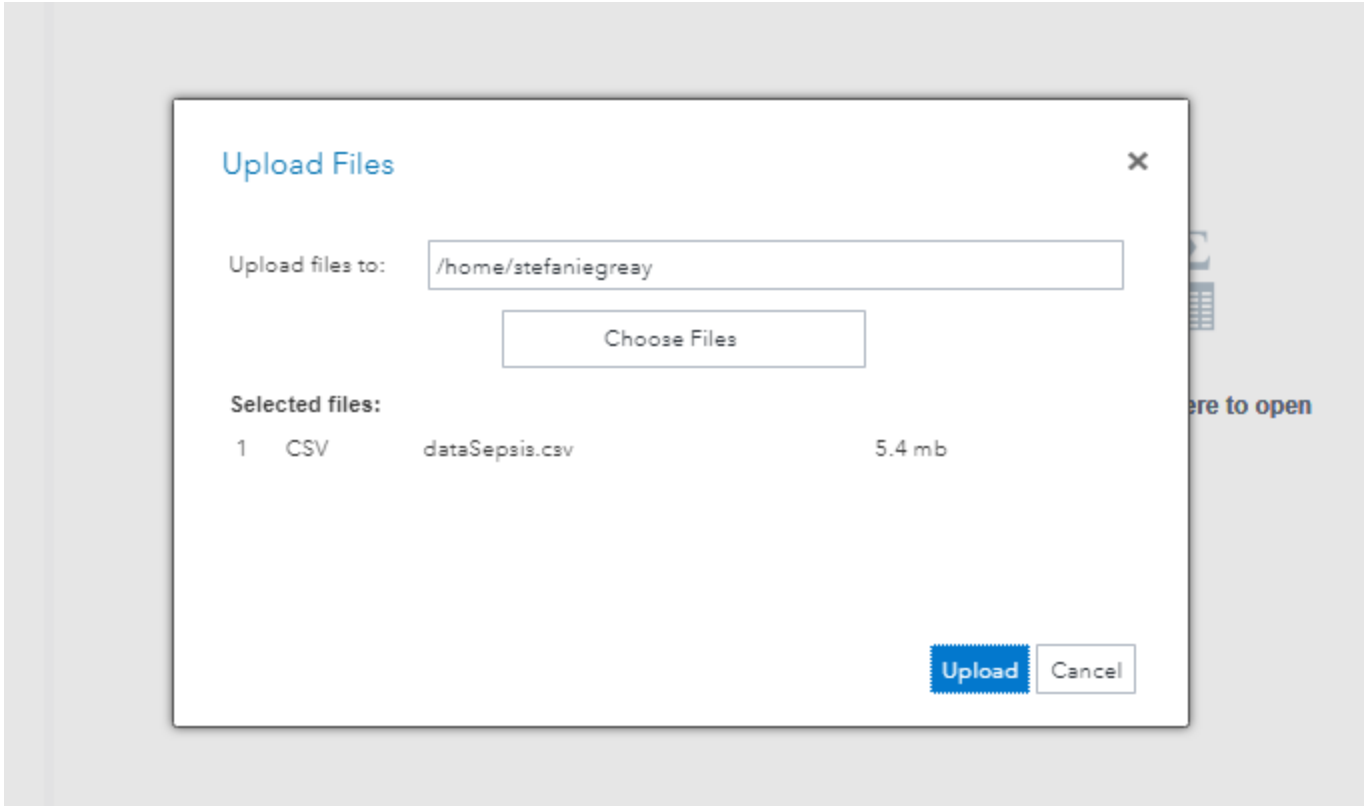
The Upload button will display in dark blue



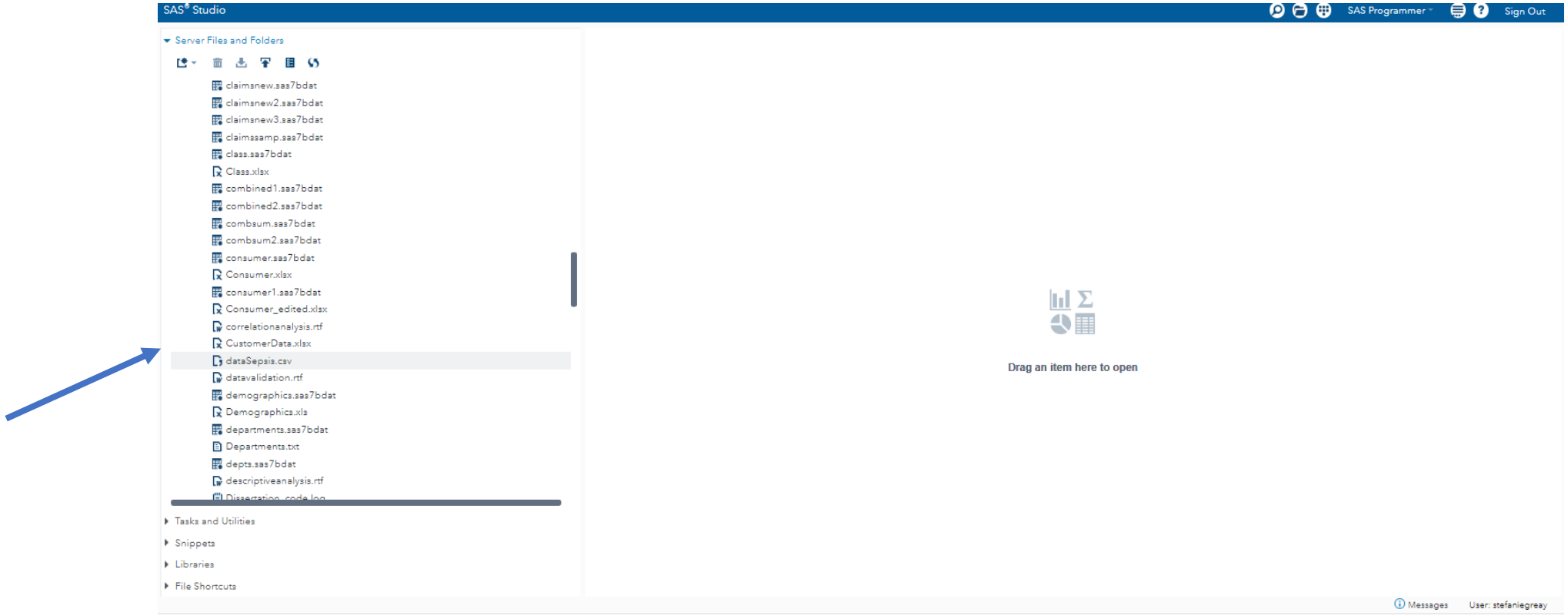
You can create a folder at this point, if you wish, or simply upload to your home directory.



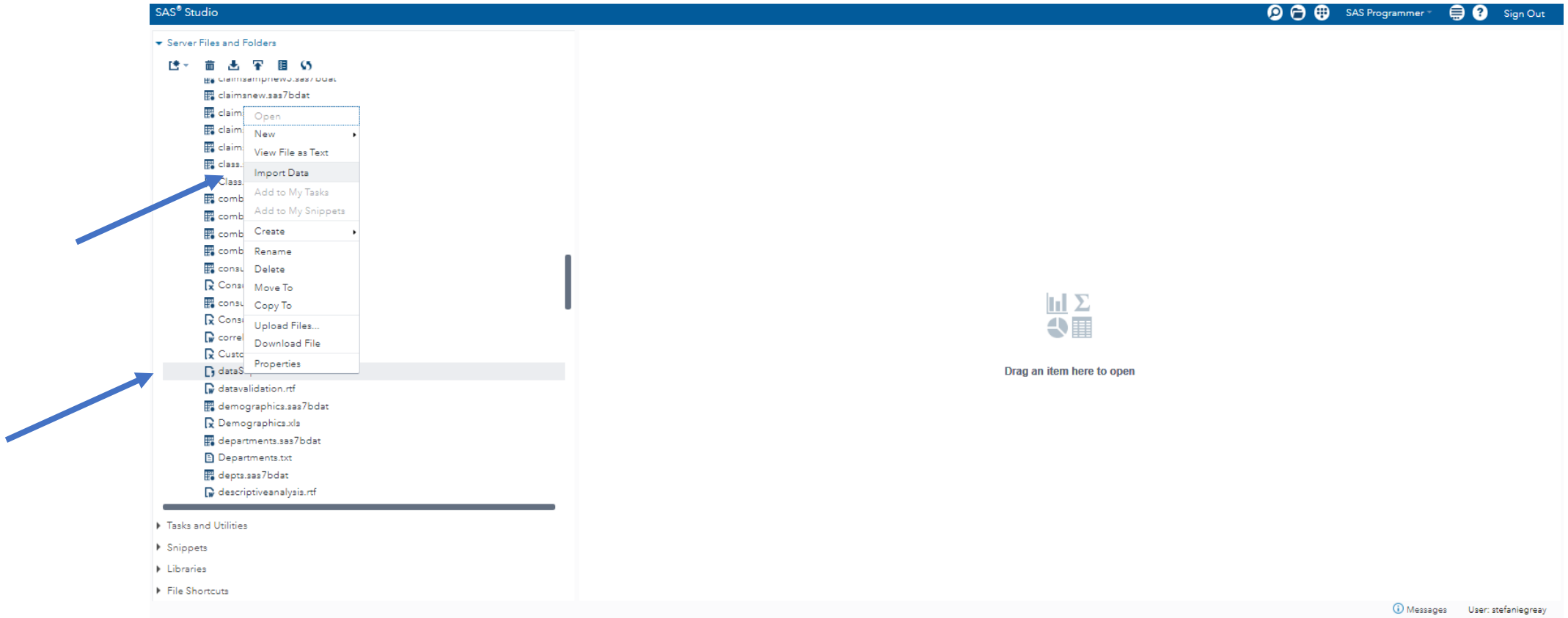
Select “Choose Files” to browse your computer for the dataset you want to upload. Once the dataset has been selected, click “Upload.”



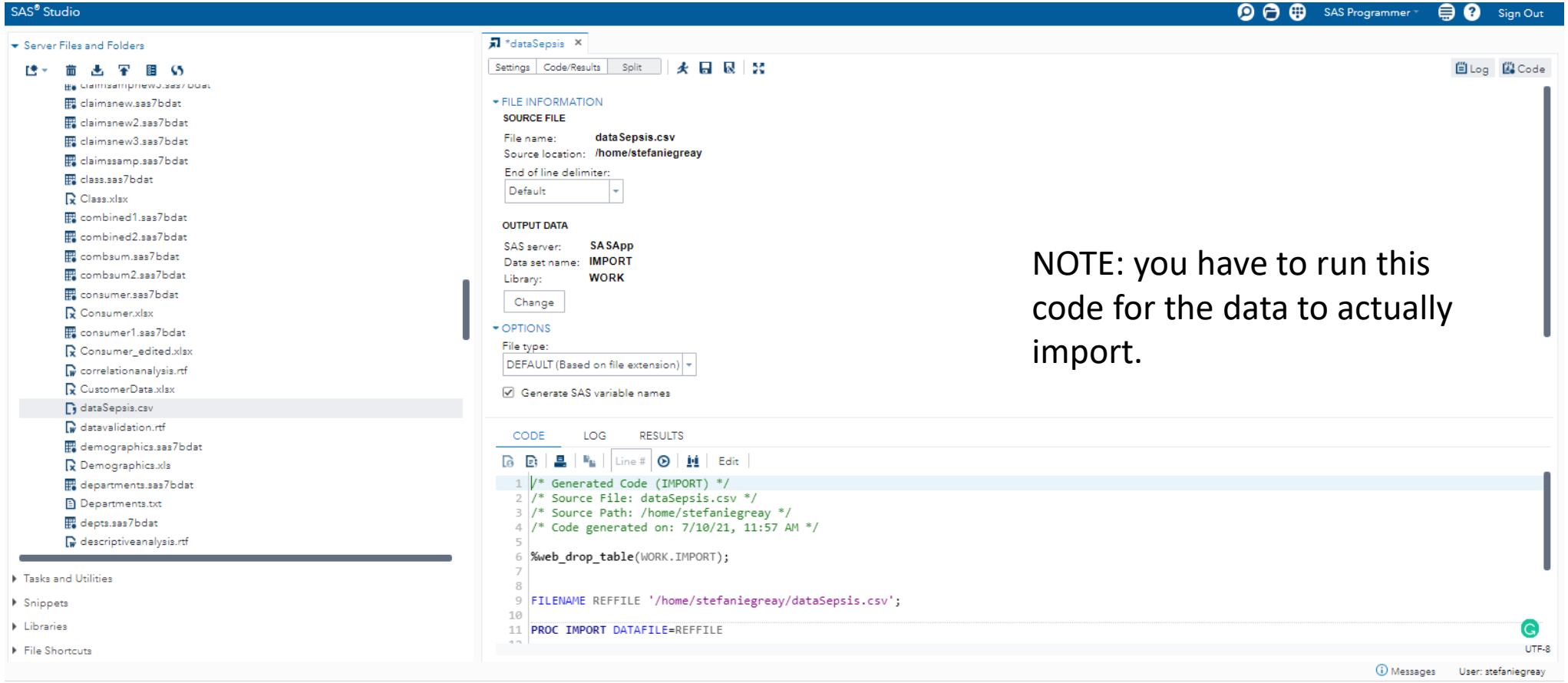
You will be able to view your files by clicking on “Files(Home)” to verify that your file successfully uploaded.



To import the dataset into a SAS dataset format (from the current csv format), right click on the name of the file, and select “Import Data.”

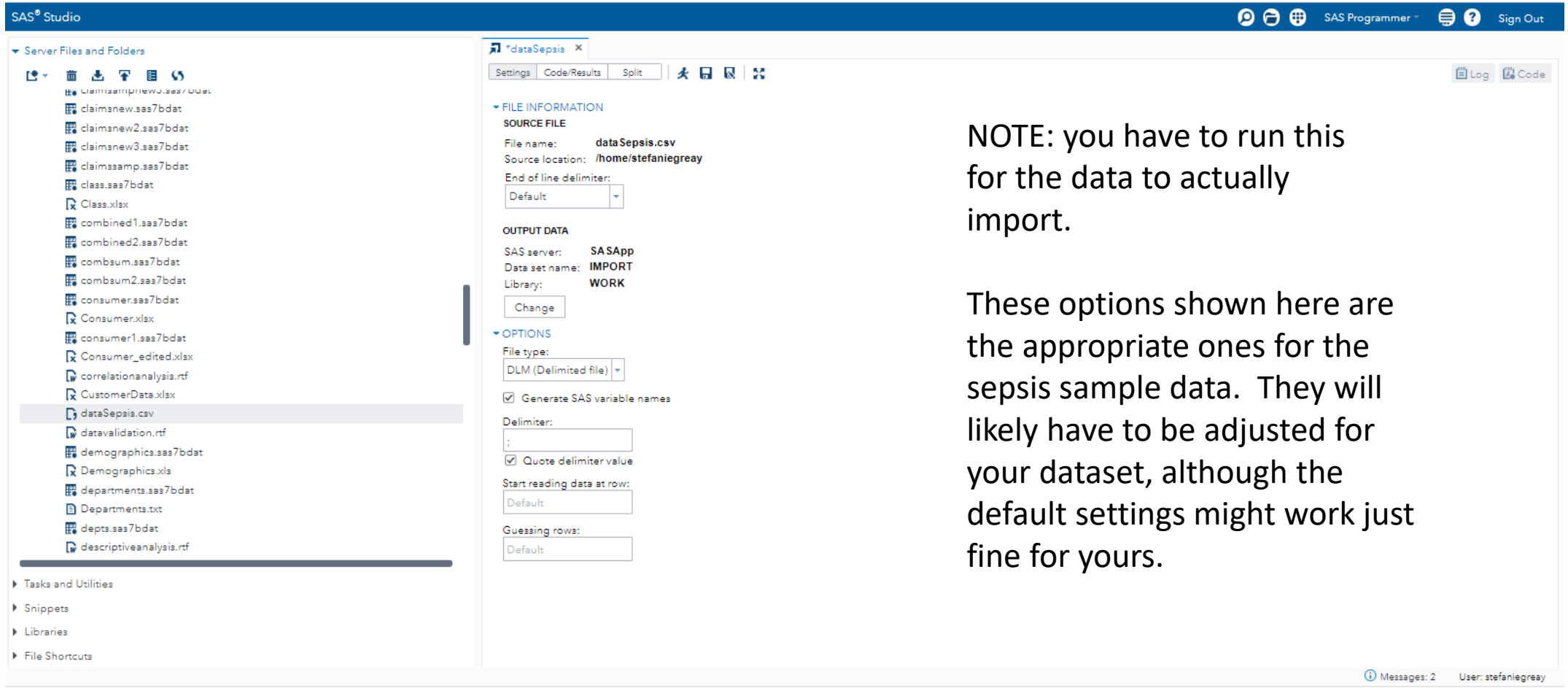


The Proc Import code will be written for you (save this as a template to use for future imports!)



NOTE: you have to run this code for the data to actually import.

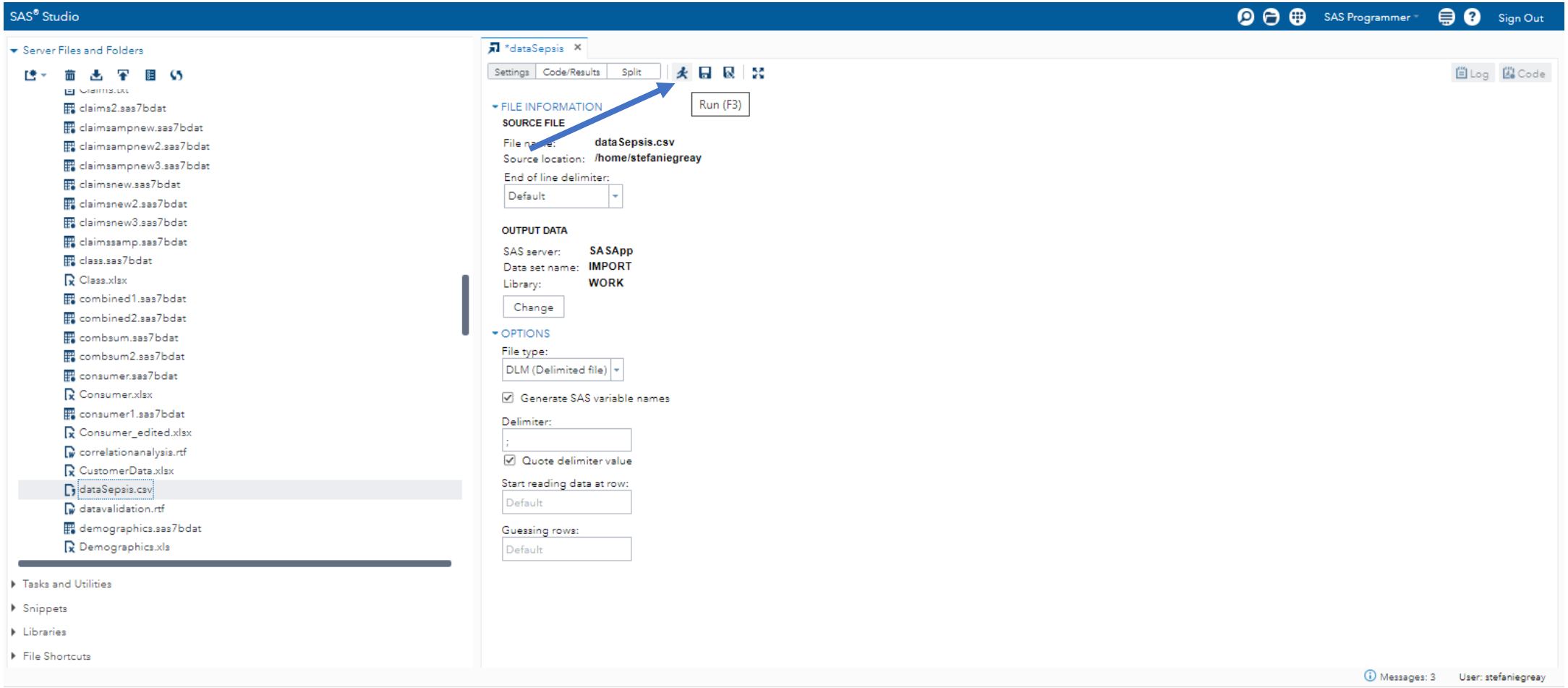
The Proc Import code will be written for you (save this as a template to use for future imports!)



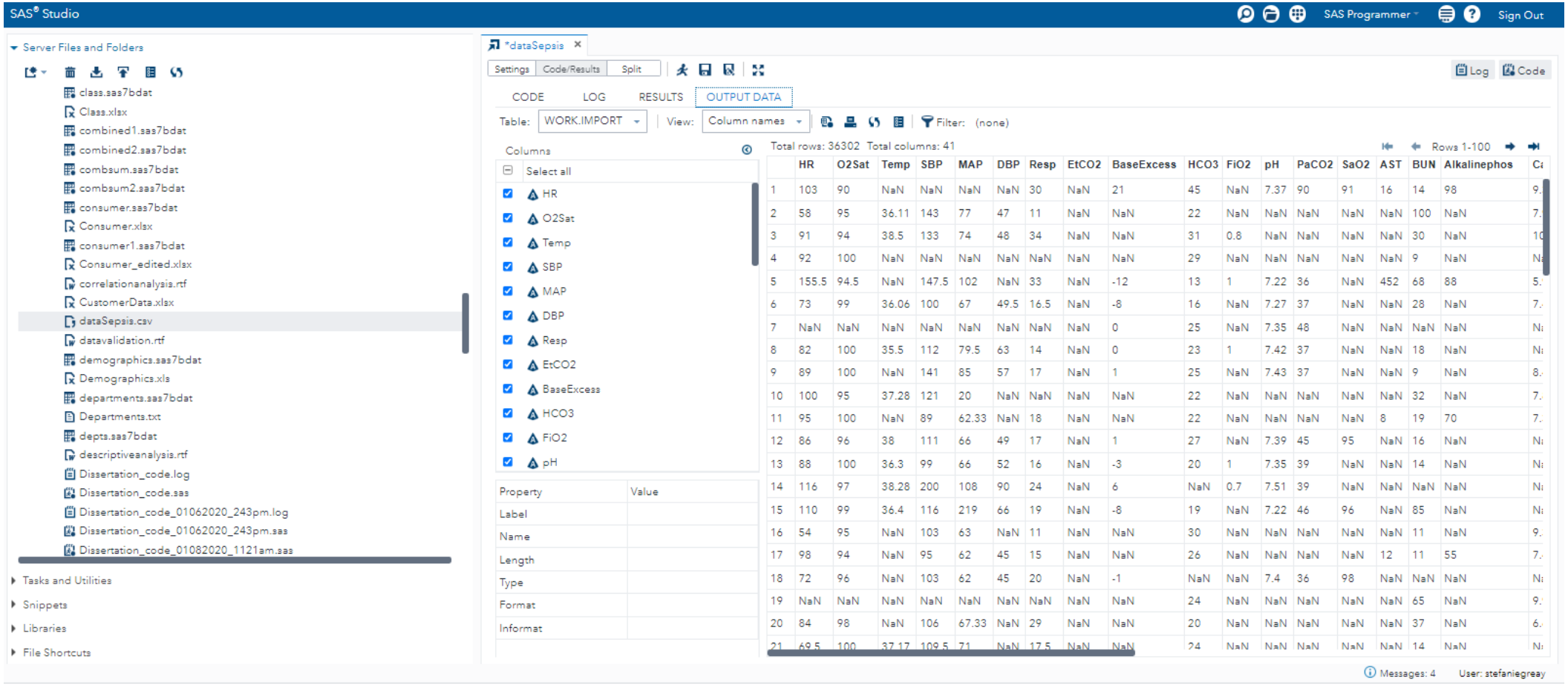
NOTE: you have to run this for the data to actually import.

These options shown here are the appropriate ones for the sepsis sample data. They will likely have to be adjusted for your dataset, although the default settings might work just fine for yours.

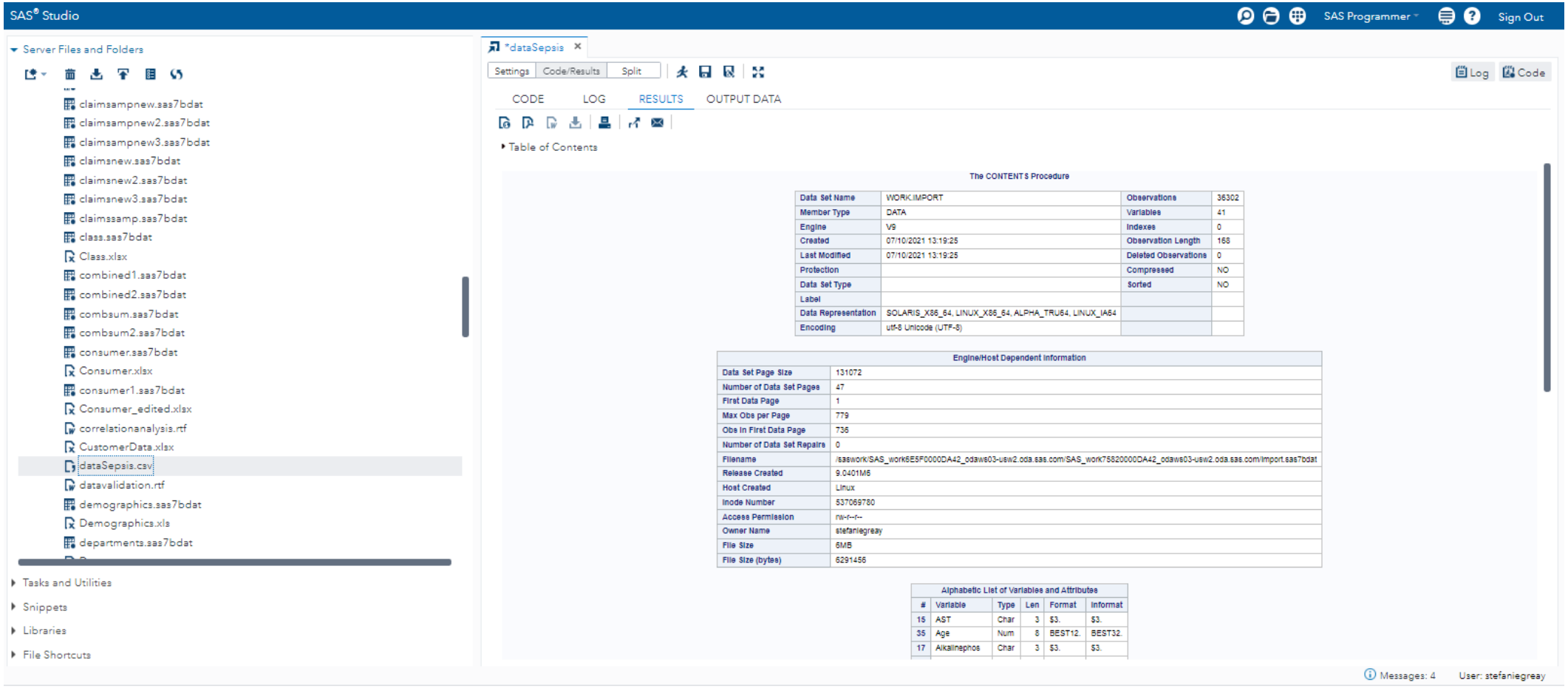
To run the code, click the icon that looks like a guy running.



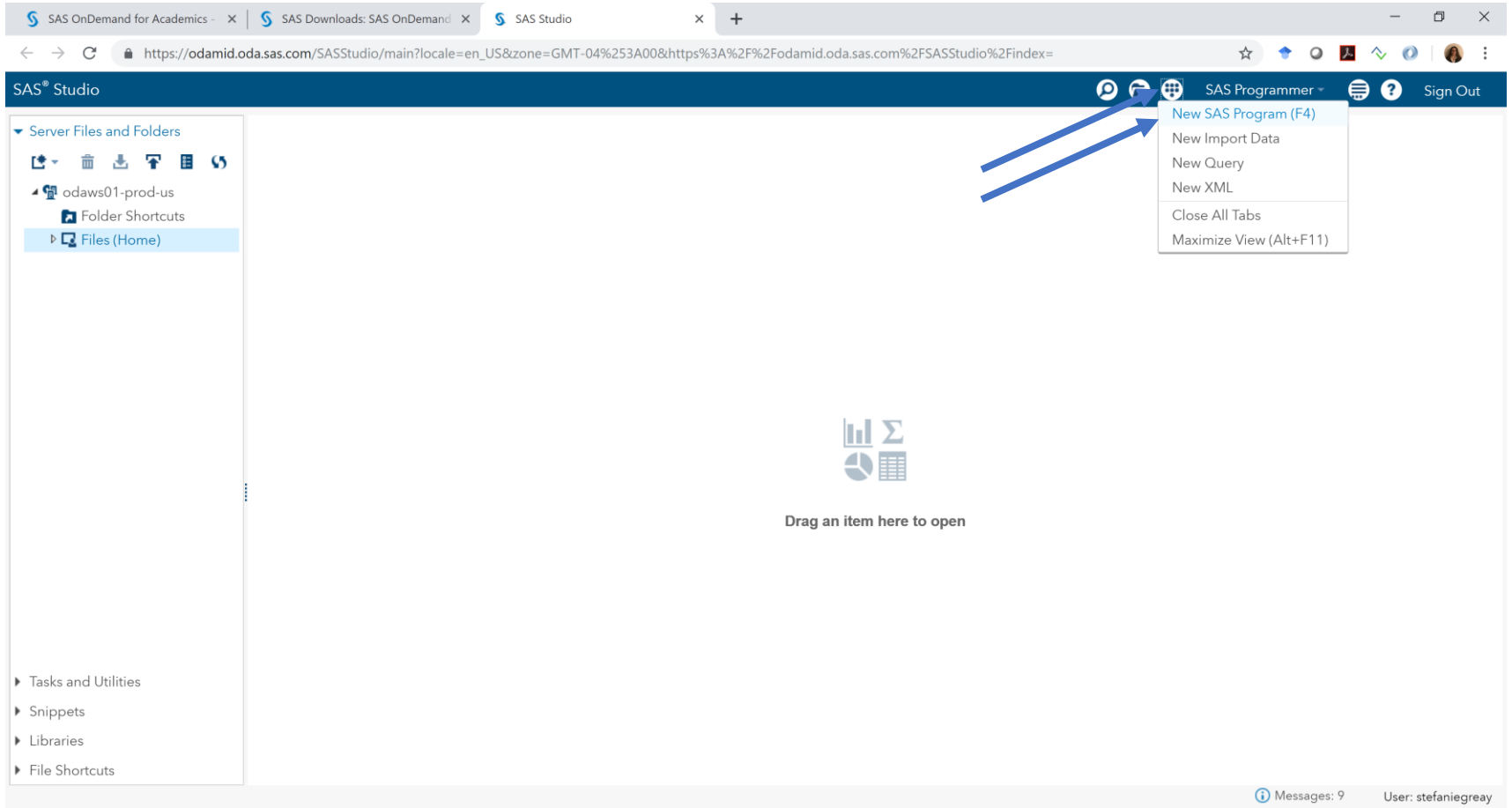
When you run the import, you will see the dataset and summary in the output data window when you click the “Code/Results” or “Split” tab and then “Output Data” and can verify its success.



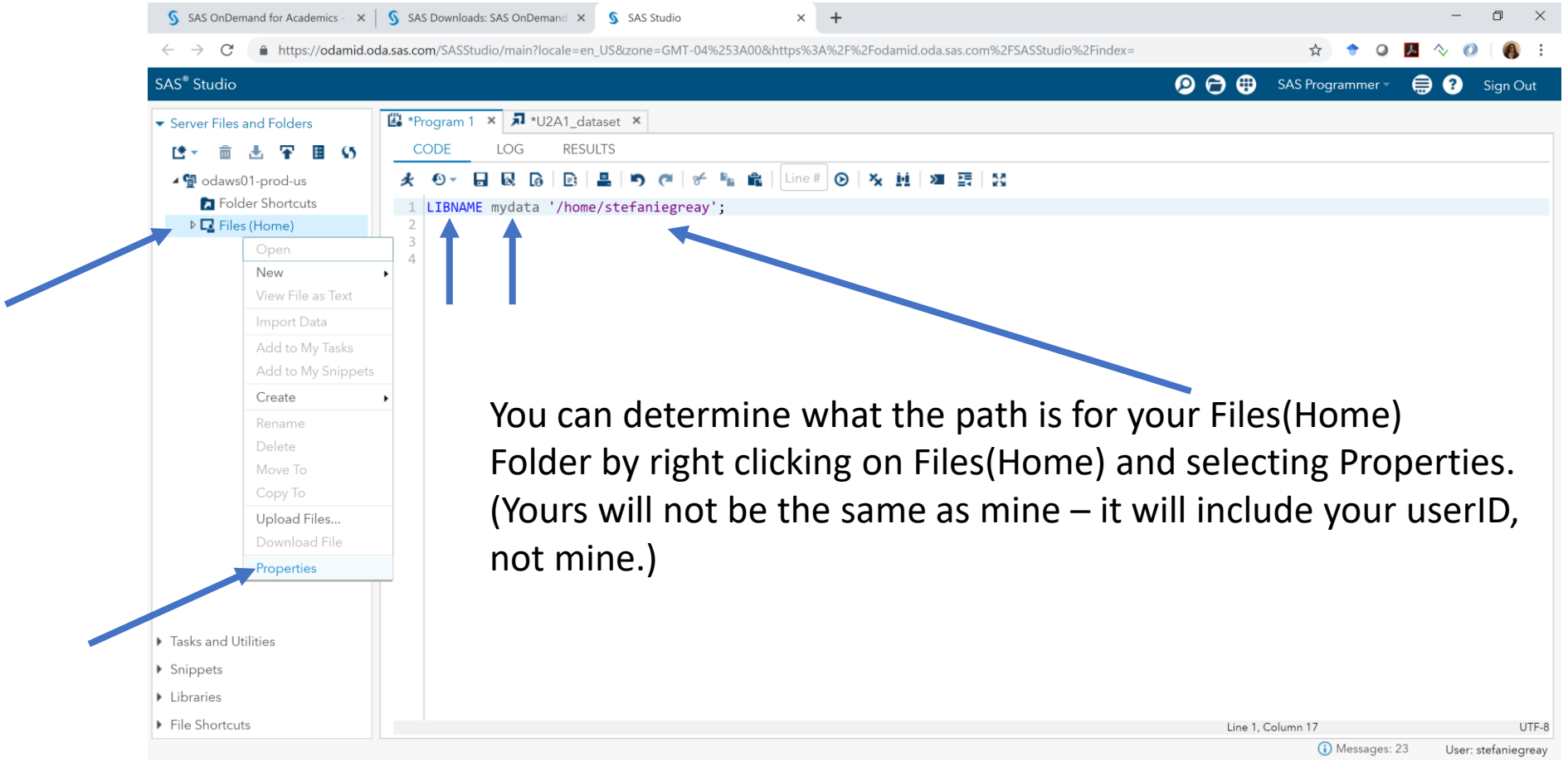
When you click the “Code/Results” or “Split” tab and then “Results,” you can see the contents of the dataset, to verify the number of observations and variables are as expected.



To get started working with the dataset you just imported, start a new SAS program.

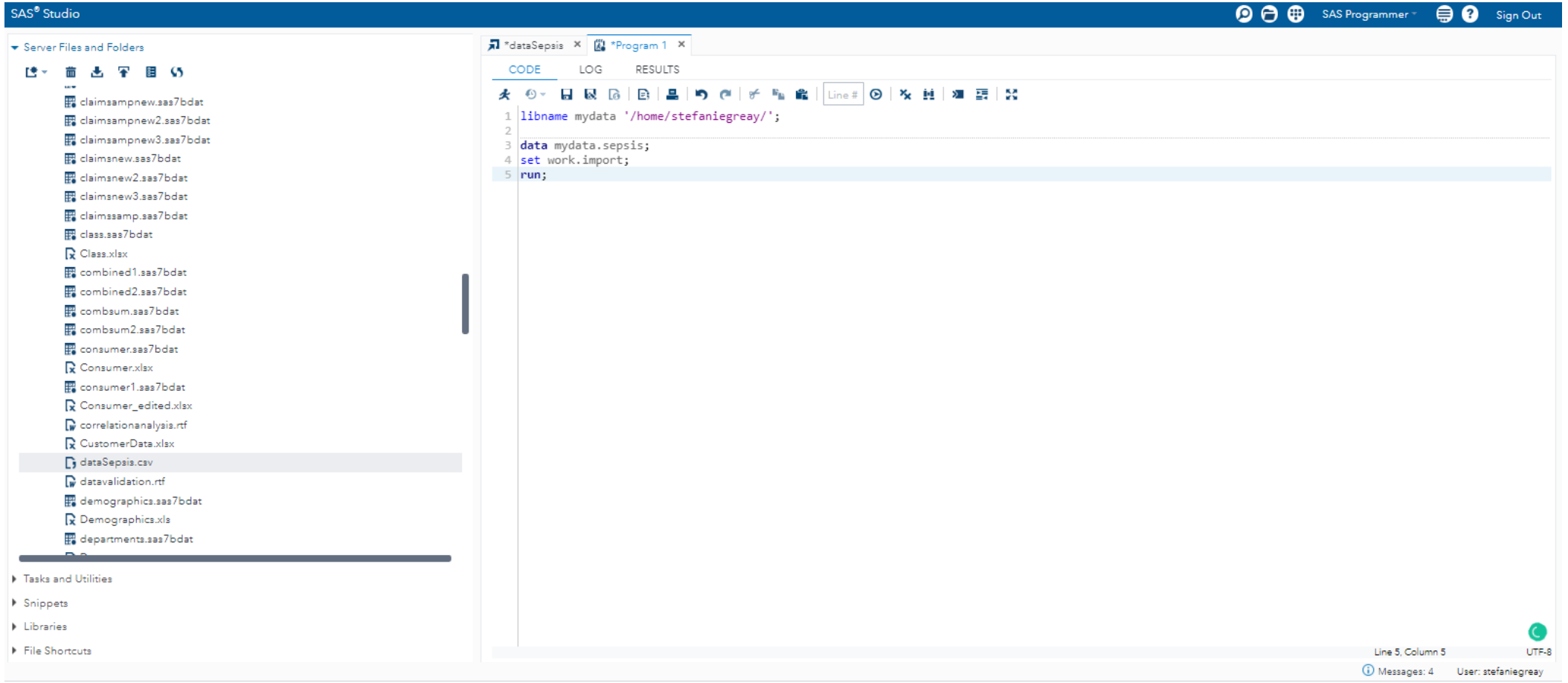


To create a SAS Library for your Files(Home) folder, you need to use a libname statement

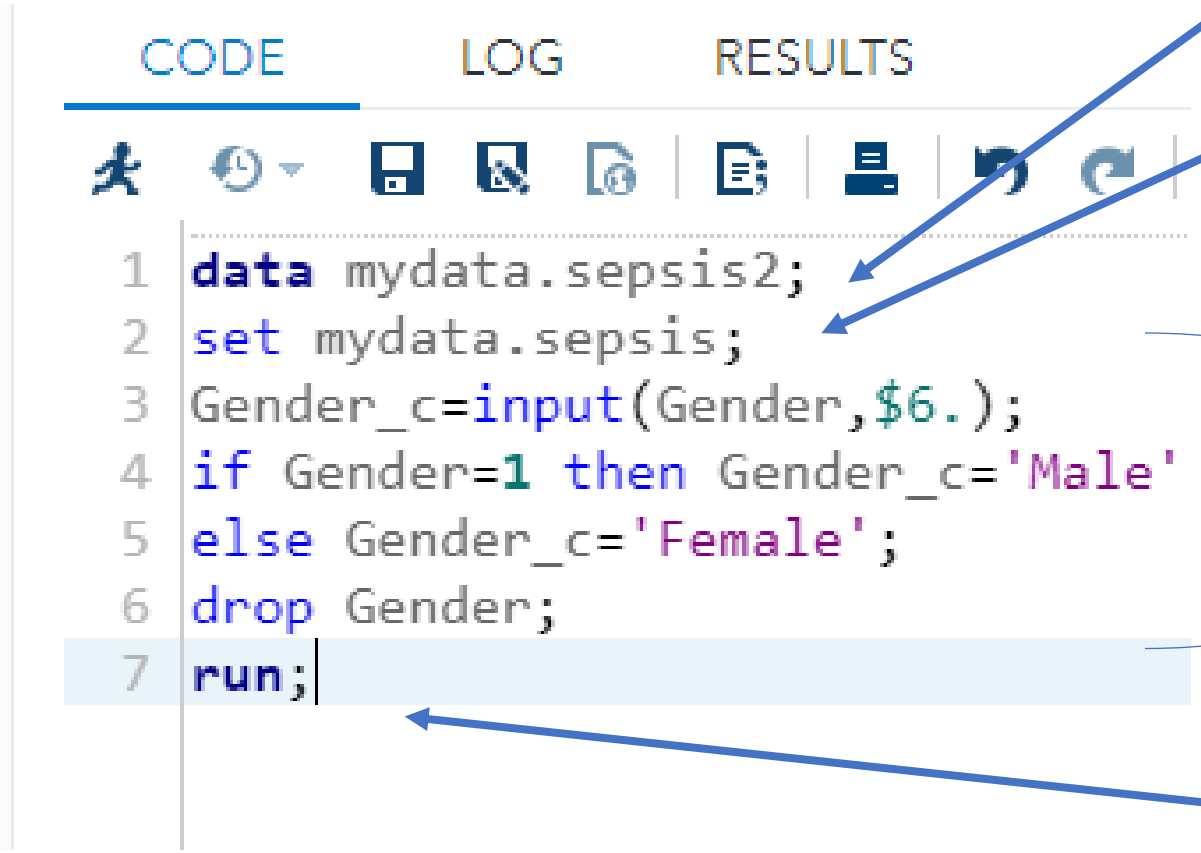


You can determine what the path is for your Files(Home) Folder by right clicking on Files(Home) and selecting Properties. (Yours will not be the same as mine – it will include your userID, not mine.)

Save the temporary SAS dataset created by the import to your library using the following sample code.



Data Step



Data statement:
Specifies the output library and dataset

Set statement:
Specifies the input library and dataset

Data manipulation statements:
These statements are where the data can be formatted, reformatted, calculated, replaced, manipulated, etc.

Run statement:
All procedures and data steps in SAS end with a run statement.

Selected Data Manipulation Examples

- Formatting a variable
- Converting a numeric variable to a character variable
- Converting a character variable to a numeric variable
- Creating a calculated variable
- Populating values of a variable using an if-then statement
- Combining multiple character variables
- Taking only a portion of a character variable using a substring

Formatting a variable

```
4 format Unit2 2.;
```

format function:
Tells SAS that it to format a specified variable using the specified format

Variable to be formatted

Format:
Tells SAS how to format the variable.
2. is a numeric format that is exactly 2 places long (specified by the 2) and the period ends the format (the period is on all informats and formats)

Converting a numeric variable to a character variable

```
4 Gender_c=input(Gender,$6.);
```

New variable name

Input function:
Tells SAS that it will be importing the values from one variable to another variable, using the informat specified

Old variable name

Informat:
Tells SAS what format to import the values of the variable in as.
$6. is a character format (specified with the $) that is 6 characters long (specified by the 6) and the period ends the format (the period is on all informats and formats)

Converting a character variable to a numeric variable

```
Unit2_n=input(Unit2,best12.);
```

New variable name

Input function:
Tells SAS that it will be importing the values from one variable to another variable, using the informat specified

Old variable name

Informat:
Tells SAS what format to import the values of the variable in as.
best12. is a numeric format that is up to 12 places long (specified by the 12) but uses the best format (specified by the best) and the period ends the format (the period is on all informats and formats)

Creating a calculated variable

New variable name

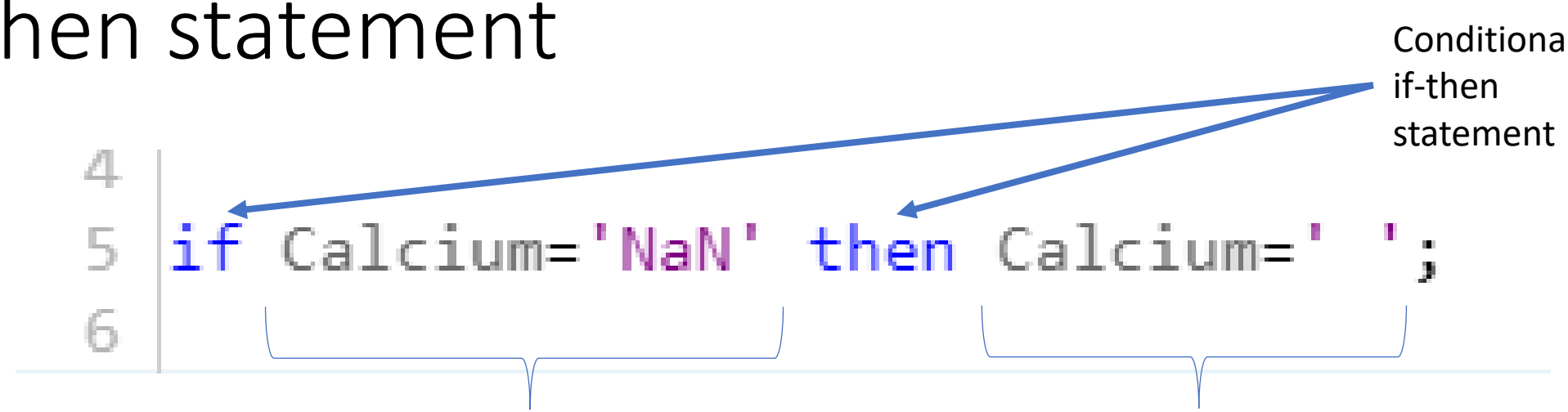
Variable 1 to be added

Mathematical operator (This can be any usual mathematical operator (+ - / * ^ for a few examples)

Variable 2 to be added

*Calculations can contain multiple variables and multiple operators and functions enclosed in parentheses. See SAS's technical documentation for additional function options that can be used in calculations.

Populating values of a variable using an if-then statement



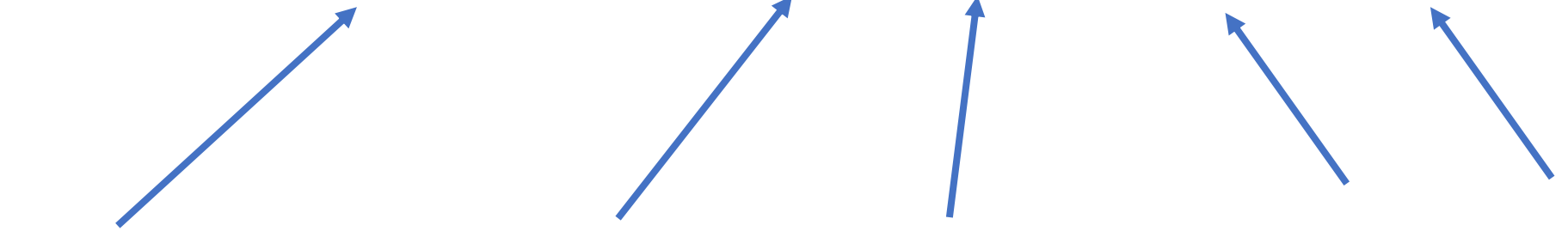
Condition:
What has to be true for the following statement to execute

Result:
What will be executed if the previous condition is met

*Note that in the sepsis dataset, null values are represented by NaN by default, but in SAS, null values in character variables are represented by a blank (like that enclosed in single quotes above) and null values in numeric variables are represented by a . (i.e. a period). So this if-then statement replaces all NaN values with the correct SAS null value of a blank.

Combining Multiple Character Variables

```
66 UnitpairID = catx(',', Unit1, Unit2);
```



New variable name

Concatination function

Separator (the symbol or character that will be used to separate the values within the new variable)

Variables to be combined

* Concatinated variables can contain two or more other variables. Note that a length statement might need to preceed the catx statement in order to ensure that the length of the new variable is long enough to include all of the concatenated values/variables.

Taking Only a Portion of a Character String

```
68 subUnitPairID=substr(UnitpairID,1,2);
```